Multimedia Signal Processing 1st Module and Fundamentals of Multimedia Signal Processing

date: November 4th, 2023

Ex.1 (Pt.12)

A Digital filter has the following difference equation:

$$y[n] = x[n] + \sqrt{2}x[n-1] + x[n-2] - 0.7 \cdot \sqrt{2}y[n-1] - 0.49y[n-2].$$

[4 pts] Provide the z transform of the filter and the pole-zero plot.

[4 pts] Depict an approximated representation of amplitude and phase response

[4 pts] The signal $x(t) = 10 + 5\cos(2\pi 50t) + 7\sin(2\pi 75t)$ is sampled at 200Hz and then filtered with the previous filter. What will be the amplitude of the spectral components of the output?

Comment the results.

Ex.2 (Pt.10)

A signal $x[n] = \{2,2,3,-1\}$ has to be filtered with the filter $h = \{1,0,-1\}$

[1 pt] Find the output signal $y[n]$ working in the time domain.

We need to process the filter in real-time adopting the Overlap and Save approach and working in the frequency domain with blocks of 4 samples.

[2 pts] Define the **W** matrix to retrieve obtain the DFT of the signal and the filter

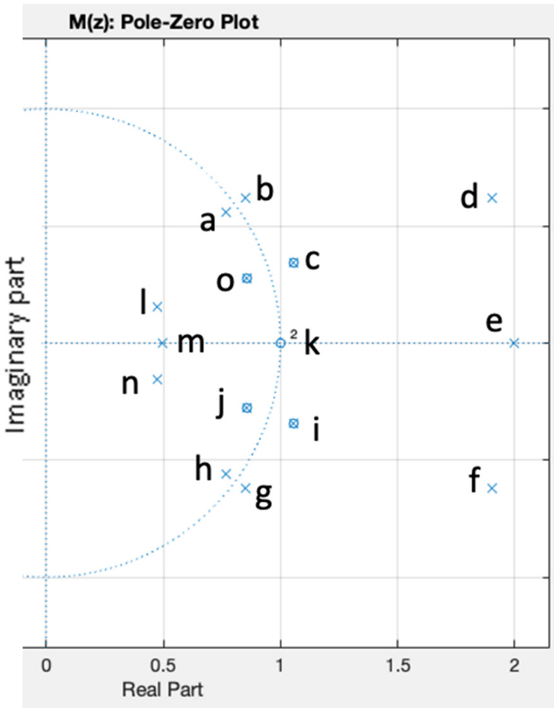
[3 pts] Define the procedure to apply the Overlap and Save to this specific context and the countermeasures to avoid unwanted effects of the circular convolution.

[4 pts] Report all the steps and the intermediate results of the Overlap and Save procedure in order to get the same result $y[n]$ of the first point.

CONTINUES ON THE BACK

Ex.3 (Pt.12) To be solved writing the MATLAB code on the sheet.

1) [3 pt] The signal x(n) repeats periodically every 20 milliseconds and it is sampled every 1 milliseconds. x(n) is composed of two cosinusoidal contributions with frequencies f0 and f1 = f0/2. Define the signal x(n) such that its DFT shows exactly 4 discrete pulses over one frequency period and that x(n) has at least 150 samples.
2) [5 pt] You are given a magnitude squared function M(z) with the following zeros-poles plot:



In particular, the root positions are:

a = $0.95e^{j\frac{\pi}{5}}$, b = $1.05e^{j\frac{\pi}{5}}$, c = $1.11e^{j\frac{\pi}{10}}$, $d = 2e^{j\frac{\pi}{10}}$,

$e = 2$, $f = 2e^{-j\frac{\pi}{10}}$, $g = 1.05e^{-j\frac{\pi}{5}}$, h = $0.95e^{-j\frac{\pi}{5}}$,

$i = 1.11e^{-j\frac{\pi}{10}}$, $j = 0.9e^{-j\frac{\pi}{10}}$, $k = 1$, l = $0.5e^{j\frac{\pi}{10}}$, $m = 0.5$, $n = 0.5e^{-j\frac{\pi}{10}}$, $o = 0.9e^{j\frac{\pi}{10}}$.

(hints: in c, i, j, o there are both zeroes and poles; we don't have roots with negative real part; all rational numbers are rounded by using two decimal digits).

- Select the correct roots to define a causal stable real-valued all-pass filter H_ap(z). Define the filter H_ap(z).
- Select the correct roots to define a causal stable real-valued minimum phase system H_min(z) such to enhance the contribution in f0 and attenuate that in f1, and such that |H_min(f=0)| = 4. Define the filter H_min(z).
- Define the filter H(z), whose all-pass/minimum-phase decomposition is defined by filters H_ap(z) and H_min(z).

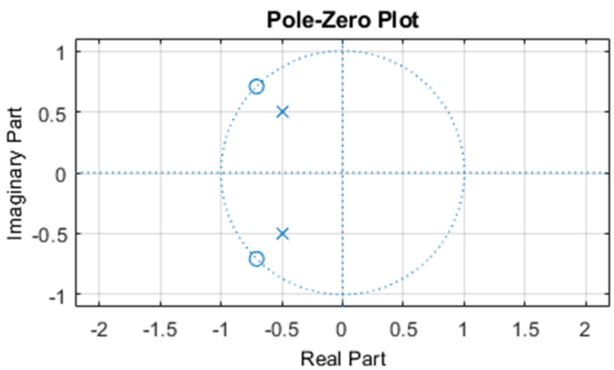
3) [4 pt] Filter the signal x(n) with the three different filters, defining the signals y_ap(n), y_min(n) and y(n).
   - Compute the DFTs of the signals y_ap(n), y_min(n) and y(n).
   - Plot the absolute values of Y_ap(f), Y_min(f) and Y(f) as a function of the normalized frequency axis starting from 0. Comment on the position/amplitude of the peaks you expect to see for every signal.
   - Provide some comments on the phase behaviour of Y_ap(f) and Y(f). Do you expect any phase jumps? Why?

Solutions

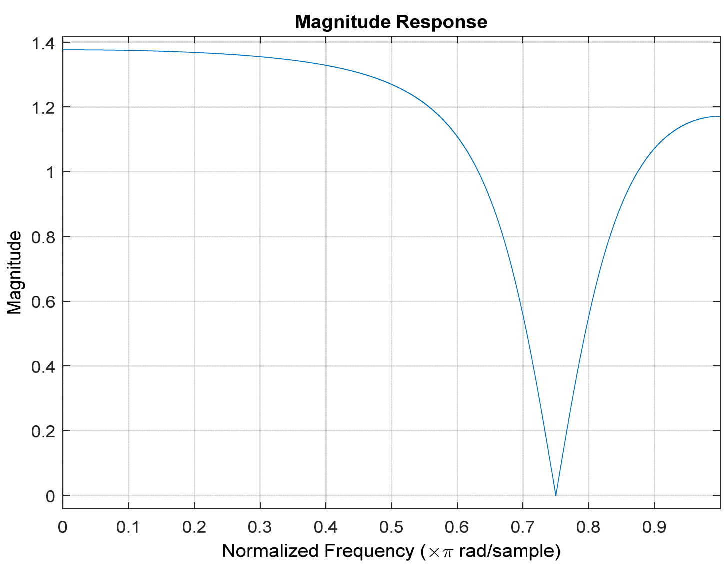
Ex.1

The z transform is: $H(z)=\frac{1+\sqrt{2}z^{-1}+z^{-2}}{1+0.7\cdot\sqrt{2}z^{-1}+0.49z^{-2}}$

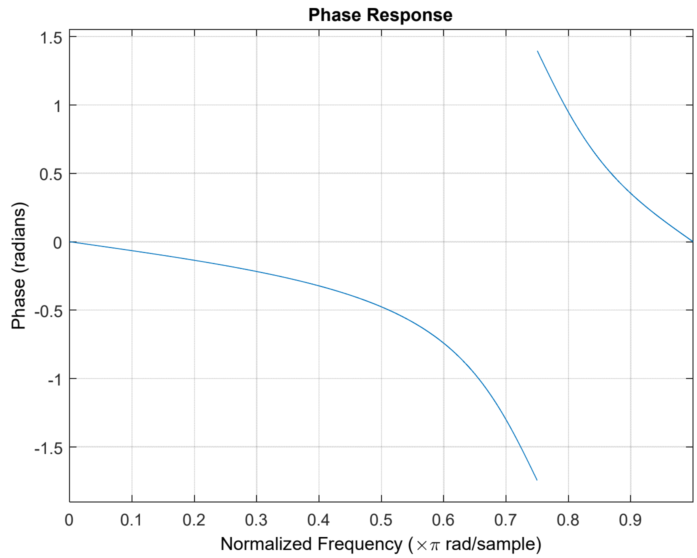
The pole-zero plot is the following:



The amplitude response is:



The phase response is:



The jump of π degrees is clearly visible at the zero.

The sampled signal is:

$$x[n]=10+5\cos\left(2\pi\frac{50}{200}n\right)+7\sin\left(2\pi\frac{75}{200}n\right)=$$
$$=10+5\cos\left(\frac{\pi}{2}n\right)+7\sin\left(\frac{3}{4}\pi n\right)$$

In order to retrieve the amplitudes of the output we have to consider the 3 components at $\omega=0,\omega=\frac{\pi}{2},\omega=\frac{3}{4}\pi$.

$$H(\omega=0\rightarrow z=1)=\left|\frac{1+\sqrt{2}+1}{1+0.7\sqrt{2}+0.49}\right|\cong 1.38$$

$$H\left(\omega=\frac{\pi}{2}\rightarrow z=j\right)=\left|\frac{1-\sqrt{2}j-1}{1-0.7\sqrt{2}j-0.49}\right|\cong 1.27$$

$$H\left(\omega=\frac{3\pi}{4}\rightarrow z=e^{j\frac{3}{4}\pi}\right)=0 \text{ due to the zero on the unit circle}$$

Ex.2

The linear convolution will give: $y[n]=\{2,2,1,-3,-3,1,0\}$

The **W** matrix will be: $W=\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -i & -1 & i\\ 1 & -1 & 1 & -1\\ 1 & i & -1 & -i\end{bmatrix}$

The DFT of the filter is:

$$H[k]=\begin{bmatrix}0\\2\\0\\2\end{bmatrix}=\begin{bmatrix}1&1&1&1\\1&-j&-1&j\\1&-1&1&-1\\1&j&-1&-j\end{bmatrix}\begin{bmatrix}1\\0\\-1\\0\end{bmatrix}$$

Since the filter $h(n)$ has a length of 3 samples the circular convolution will give a circular tail of 3-1=2 samples.

To process the first block $x_1[n]$ of the input signal we must add 2 zeros before the first sample: the result will be:

$$X_1[k]=\begin{bmatrix}1&1&1&1\\1&-j&-1&j\\1&-1&1&-1\\1&j&-1&-j\end{bmatrix}\begin{bmatrix}0\\0\\2\\2\end{bmatrix}=\begin{bmatrix}4\\-2+2j\\0\\-2-2j\end{bmatrix}$$

$$Y_1[k]=X_1[k]\cdot H[k]=\begin{bmatrix}0\\-4+4j\\0\\-4-4j\end{bmatrix}$$

$$y_1[n]=W^{-1}Y_1[k]=\frac{1}{4}\begin{bmatrix}1&1&1&1\\1&j&-1&-j\\1&-1&1&-1\\1&-j&-1&j\end{bmatrix}\begin{bmatrix}0\\-4+4j\\0\\-4-4j\end{bmatrix}=\begin{bmatrix}{\color{red}-2}\\{\color{red}-2}\\2\\2\end{bmatrix}$$

The first two samples (in red) will be thrown away due to the unwanted effect of circular convolution.

Then I will restart the processing on the second block from the 1st sample of the input sequence:

$$X_2[k]=\begin{bmatrix}1&1&1&1\\1&-j&-1&j\\1&-1&1&-1\\1&j&-1&-j\end{bmatrix}\begin{bmatrix}2\\2\\3\\-1\end{bmatrix}=\begin{bmatrix}6\\-1-3j\\4\\-1+3j\end{bmatrix}$$

$$Y_2[k]=X_2[k]\cdot H[k]=\begin{bmatrix}0\\-2-6j\\0\\-2+6j\end{bmatrix}$$

$$y_2[n] = W^{-1} Y_2[k] = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}\begin{bmatrix} 0 \\ -2-6j \\ 0 \\ -2+6j \end{bmatrix} = \begin{bmatrix} -1 \\ 3 \\ 1 \\ -3 \end{bmatrix}$$

Again, the first two output samples must be thrown away.

The we have to repeat the procedure for the last two samples with thethird block starting from the 3rd sample:

$$X_3[k] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}\begin{bmatrix} 3 \\ -1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 3+j \\ 4 \\ 3-j \end{bmatrix}$$

$$Y_3[k] = X_3[k] \cdot H[k] = \begin{bmatrix} 0 \\ 6+2j \\ 0 \\ 6-2j \end{bmatrix}$$

$$y_3[n] = W^{-1} Y_3[k] = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}\begin{bmatrix} 0 \\ 6+2j \\ 0 \\ 6-2j \end{bmatrix} = \begin{bmatrix} 3 \\ -1 \\ -3 \\ 1 \end{bmatrix}$$

Collecting the useful portions (black ones) of the results we obtain the same result of the linear convolution.

Ex.3 (MATLAB CODE)

```
close all
clearvars
clc

%% 1. [3 pt]

% The signal x(n) repeats periodically every 20 milliseconds and it is
% sampled every 1 milliseconds. x(n) is composed of two cosinusoidal
% contributions with frequencies f0 and f1 = f0/2. Define the signal
% x(n) such that its DFT shows exactly 4 discrete pulses over one
% frequency period and that x(n) has at least 150 samples.

P = 20e-3;
Ts = 1e-3;

% To find f0 and f1 frequencies, we need to consider that the overall
% period of x(n) is the least common multiple between the two periods.
% Let's impose this by reasoning in number of samples
P_samples = P/Ts;
```

```
% P0_samples = 1/f0_norm;
% P1_samples = 1/f1_norm = 2/f0_norm = 2*P0_samples
% --> the least common multiple between P0_samples and 2*P0_samples =
% 2*P0_samples --> we impose it to be = P_samples and we find f0_norm.
f0_norm = 2/P_samples;
f1_norm = f0_norm/2;

% To have exactly four deltas in the DFT, the number of signal samples
% should be a multiple of the period.
% If we need to consider at least 150 samples, let's find the nearest
% integer multiple of the period which is greater or equal to 150.
min_sample_number = 150;
% put this number in ratio with P_samples
num_periodic_repetitions = ceil(min_sample_number/P_samples);
% number of samples
N = P_samples * num_periodic_repetitions;

% define the discrete sample temporal axis (from 0 to N-1)
n_axis = 0: N - 1;

% signal definition
x = cos(2*pi*f0_norm*n_axis) + cos(2*pi*f1_norm*n_axis);

%% 2. [5 pt]

% You are given a magnitude squared function M(z) with the following
% zeros-poles plot (check the text for the root values).
% Select the correct roots to define a causal stable real-valued all-
pass
% filter H_ap(z). Define the filter H_ap(z).
% Select the correct roots to define a causal stable real-valued
% minimum phase system H_min(z) such to enhance the contribution in
% f0 and attenuate that in f1, and such that |H_min(f=0)| = 4.
% Define the filter H_min(z).
% Define the filter H(z), whose all-pass/minimum-phase decomposition
% is defined by filters H_ap(z) and H_min(z).

% All-pass filter
% The only zero-pole combination which returns a causal stable real-
valued
% all-pass filter is: select poles in o and j; select zeros in c and
i.
% (zeroes and poles should be in conjugate reciprocal pairs)
poles = [.9*exp(1i*pi/10); .9*exp(-1i*pi/10)];
zeroes = 1./conj(poles);
B_ap = poly(zeroes);
A_ap = poly(poles);
% remember to adjust the gain such that |H_ap(f)| = 1 for each f
% we find c0 by substituting z = 1 (which corresponds to f = 0) in the
% polynomials
c0 = sum(A_ap)/sum(B_ap);
B_ap = B_ap * c0;

% to better analyze the filter (not required)
[H_ap, omega] = freqz(B_ap, A_ap, 1024, 'whole');
```

```
figure,
plot(omega./(2*pi), abs(H_ap));
title('|DTFT| of the filter H_{ap}(z)');
grid;
xlabel('f [norm]');

% Minimum-phase filter
% f0 corresponds to omega_0_norm = pi/5
% f1 corresponds to omega_1_norm = pi/10
% zeros and poles should be inside the unit circle.
% to attenuate f1, select zeros in o and j.
% to enhance f0, select poles in a and h.
poles = [0.95*exp(1i*pi/5); 0.95*exp(-1i*pi/5)];
zeroes = [.9*exp(1i*pi/10); .9*exp(-1i*pi/10)];
B_min = poly(zeroes);
A_min = poly(poles);
% the gain in f = 0 should be = 4.
c0 = 4 * sum(A_min)/sum(B_min);
B_min = B_min * c0;

% to better analyze the filter (not required)
[H_min, omega] = freqz(B_min, A_min, 1024, 'whole');
figure,
plot(omega./(2*pi), abs(H_min));
title('|DTFT| of the filter H_{min}(z)');
grid;
xlabel('f [norm]');

% Filter H(z)
% to find it, exploit the convolution property
B = conv(B_ap, B_min);
A = conv(A_ap, A_min);

% to better analyze the filter (not required)
[H, omega] = freqz(B, A, 1024, 'whole');
figure,
plot(omega./(2*pi), abs(H));
title('|DTFT| of the filter H(z)');
grid;
xlabel('f [norm]');

%% 3. [4.5 pt]

% Filter the signal x(n) with the three different filters,
% defining the signals y_ap(n), y_min(n) and y(n).
% Compute the DFTs of the signals y_ap(n), y_min(n) and y(n).
% Plot (with the stem function) the absolute values of Y_ap(f),
% Y_min(f) and Y(f) as a function of the normalized frequency axis
% starting from 0. Comment on the position/amplitude of the peaks
% you expect to see for every signal.
% Provide some comments on the phase behaviour of Y_ap(f) and Y(f).
% Do you expect any phase jumps? Why?

% filter the signal
y = filter(B, A, x);
```

```
y_ap = filter(B_ap, A_ap, x);
y_min = filter(B_min, A_min, x);

% DFTs
Y = fft(y);
Y_min = fft(y_min);
Y_ap = fft(y_ap);

% normalized frequency axis:
freq_axis = 0:1/N:1 - 1/N;

figure;
stem(freq_axis, abs(Y));
title('Absolute value of the DFT of the signal y(n)');
grid;
xlabel('f [norm]');
% We expect to see mainly two peaks in f0_norm and 1 - f0_norm. The
peaks
% corresponding to f1_norm have been strongly attenuated by H(z).

figure;
stem(freq_axis, abs(Y_min));
title('Absolute value of the DFT of the signal y_{min}(n)');
grid;
xlabel('f [norm]');
% We find basically no differences with respect to y(n). H(z) and
H_min(z)
% differ only for the all-pass component, which has no effect on the
% amplitude.

figure;
stem(freq_axis, abs(Y_ap));
title('Absolute value of the DFT of the signal y_{ap}(n)');
grid;
xlabel('f [norm]');
% The peaks are basically the same as x(n), because the filter is an
all-pass

% phase behaviour of Y_ap(f): we expect to see phase jumps, because
every
% all-pass filter which is stable and causal is maximum phase.
% phase behaviour of Y(f): we expect to see phase jumps as well,
because
% H(z) contains maximum-phase zeroes.
```